

**REMARKS**

This communication is in response to the first Office Action dated November 6, 2003. In that Office Action the Examiner objected to some clerical errors in the Figure 16. A new replacement sheet for Figure 16 is enclosed herewith. Further, the Examiner rejected Claims 1, 3, 4, 9, 11-12, 20-21, 23-24, 29, 32, and 34-35 as being anticipated by U.S. Patent No. 6,292,488 to Filgate. The remaining claims were rejected as being obvious over Filgate in combination with various secondary references.

**The Present Invention**

The present invention in general operates in the context of an interconnect fabric module ("IFM") that has high-speed switching capabilities. An interconnect fabric module can be dynamically configured to interconnect its communications ports so that data can be transmitted through the interconnected ports. Multiple interconnect fabric modules can be connected to form an interconnect fabric through which nodes (e.g., computer systems) can be interconnected.

[0001] The claims of the present application are directed primarily towards a method and apparatus for avoiding deadlocks between interswitches in the IFM. While the present specification comprehensively covered all aspects of an IFM, the present claimed invention is directed towards only those specific aspects related to deadlock avoidance.

[0002] Specifically, switch protocol controllers of the IFMs implement a deadlock avoidance scheme to prevent interswitch deadlocks. An interswitch deadlock may occur when two partially built connections both need the same port to complete their connections. Figure 16 is a block diagram illustrating an interswitch deadlock. In this example, node 1605 requests that a connection be established to node 1607, and node 1606 requests that a connection be established to node 1605. Node 1605 is directly linked to port 1 of interconnect fabric module 1601 and

nodes 1606 and 1607 are directly linked to ports 1 and 2, respectively, of interconnect fabric module 1602.

[0003] Table 1610 illustrates a sequence of events that results in a deadlock. At time 0, nodes 1605 and 1606 send out start-of-connection frames. At time 1, the interconnect fabric module 1601 establishes a crosspoint connection between its port 1 and its port 0 via crosspoint switch 1603 as part of the process of establishing the connection between node 1605 and node 1607. At the same time, interconnect fabric module 1602 establishes a crosspoint connection between its port 1 and port 0 via crosspoint switch 1604 as part of the process of establishing the connection between node 1606 and node 1605. At time 2, interconnect fabric module 1601 transmits the start-of-connection frame to interconnect fabric module 1602 via link 1608, and interconnect fabric module 1602 transmits the start-of-connection frame to interconnect fabric module 1601 via link 1608. When interconnect fabric module 1601 receives the start-of-connection frame, it determines that it cannot currently establish a crosspoint connection from port 0 to port 1 because port 0 is in use by the partial connection from node 1605 to node 1607.

[0004] Similarly, when interconnect fabric module 1602 receives the start-of-connection frame, it determines that it cannot currently establish a crosspoint connection from port 0 to port 2 because port 0 is in use by the partial connection from node 1606 to node 1605. Because neither connection can be completed, a deadlock occurs.

[0005] A switch protocol controller uses an interswitch deadlock avoidance scheme. Whenever a switch protocol controller receives a start-of-connection frame and the switch protocol controller is currently in a connection, then a conflict has occurred. The switch protocol controller receives such a conflicting start-of-connection frame when the conflicting start-of-connection frame was initially transmitted from a node before the connection that included that switch protocol controller's port was complete. To avoid a deadlock, **once the conflict is**

detected, the switch protocol controller compares the priority of the conflicting start-of-connection frame with the priority of the start-of-connection frame for its partially built connection to determine which connection should be established. If the frames have the same priority, then the switch protocol controller uses the domain address identifier or other unique identifier of the interconnect frame modules as a tiebreaker, that is the interconnect fabric module that received and the one that sent the conflicting start-of-message frame. If the priority of the conflicting frame is higher, then the switch protocol controller sends a frame through its input direction indicating that the connection cannot be established and then proceeds to process the conflicting start-of-connection frame to complete the connection. Conversely, the switch protocol controller that sent the conflicting frame also detects the conflict but determines that the frame it sent has a higher priority and ignores the start-of-connection frame that it just received.

*The Filgate Reference*

The Filgate patent discloses a recovery mechanism for devices communicating across a long haul simplex data link. A computer system includes a host device and a target device at each of at least two locations. The host and target devices for a given location are connected by a local data bus segment. Each location also includes a gateway device, e.g., a communication bridge, which connects its local data bus segment to a long haul simplex data link, allowing it to communicate with other locations.

When a deadlock occurs, the deadlock recovery mechanism prompts the device which currently controls it to disconnect from the bridge. When a device disconnects, it relinquishes control of the bridge. Once control of each bridge is relinquished, each bridge no longer forwards its own request for control of the other bridge via the long haul data link. Therefore, each bridge becomes idle, which breaks the deadlock. Each device which originally controlled a bridge attempts to re-take control of the communication path. Various device and system

characteristics make it highly improbable that each device will attempt to re-take control at substantially the same time and, therefore, it is also highly improbable that the devices will immediately cause another deadlock.

Further, Filgate teaches that the deadlock recovery mechanism within each bridge uses its own unique time delay to control recovery of the system. Once a deadlock occurs, each recovery mechanism prompts its bridge to terminate its request for the other bridge via the simplex long haul data link. A device local to each bridge still has a request pending with its respective bridge to take control of the full communication path, but each recovery mechanism prevents its bridge from servicing its local device's request until the bridge's time delay has expired. The bridge with the shorter time delay, referred to as the "priority bridge", will service its device's request first and, thus, gain control of the other bridge.

*The Examiner's Arguments*

The Examiner argues that Filgate anticipates each of the independent claims. However, there is at least one fundamental difference between Filgate and the present claimed invention. Specifically, Filgate teaches that once a deadlock occurs, both **nodes drop the connection and attempt to reconnect**. Col. 4, lines 20-32; see also Col. 5, lines 47-50. This is different than the present claimed invention which determines if a "current connection has a higher priority than the conflicting connection, [and] **maintaining the current connection**". See Claim 1 (emphasis added). In other words, the claimed invention keeps a connection with one of the conflicting nodes, while a priority determination is made. The Filgate patent does not maintain a connection, but instead simply drops both connections and has them reconnect at a later time. Therefore, for this reason, Filgate does not anticipate Claim 1. The other independent claims contain similar limitations and thus are also allowable for the same reasons.

Further, independent Claims 20 and 32 have been amended to include the limitations of Claims 22 and 33, respectively. This limitation relates to having the priority be based upon the priority of the communications to be transmitted through the connection. The Examiner acknowledges that this limitation is not shown in Filgate, but instead relies upon U.S. Patent No. 5,892,923 to Yasuda. In general, Yasuda teaches nothing more than including a priority indication in a message packet. The priority information can be used by a switch to route the message packet through the network. There is no indication in Yasuda that the priority indication may be **used to control whether or not a connection should be made or be disconnected in favor of another connection.** Therefore, there is no suggestion to make a combination of Yasuda with Filgate and the Examiner's rejection is improper.

As noted above, Filgate teaches that both conflicting nodes attempting to use the same port (bridge) are disconnected. Thus, because of this, it would be impossible to combine the teachings of Yasuda with Filgate, since Yasuda teaches that the priority is embedded into the message packet. If Yasuda were combined with Filgate, the hypothetical system would never receive the priority information since Filgate disconnects any connections prior to any information transfer. For this reason, it is not obvious to combine Filgate with Yasuda.

In view of the foregoing, the claims pending in the application comply with the requirements of 35 U.S.C. § 112 and patentably define over the applied art. A Notice of Allowance is, therefore, respectfully requested. If the Examiner has any questions or believes a telephone conference would expedite prosecution of this application, the Examiner is encouraged to call the undersigned at (206) 359-6488.

Respectfully submitted,  
Perkins Coie LLP

Date: \_\_\_\_\_

*12/26/04*

\_\_\_\_\_  
Chun M. Ng  
Registration No. 36,878

Correspondence Address:  
Customer No. 25096  
Perkins Coie LLP  
P.O. Box 1247  
Seattle, Washington 98111-1247  
(206) 359-6488